

Working with `getElementsByName`

`getElementsByName()` is a method (or function) that get an array filled with all the elements in the document that have the specified tag name sent as a parameter.

For example:

```
document.getElementsByTagName('div')
```

- refers to all DIV tags.

This function stores the items into an array-object (a sequential array), where the keys are integers starting from 0.

If we wanna refer to the first <div>, we use:

```
var div1 = document.getElementsByTagName('div')[0]
```

- For the second <div>, we use index [1], and so on.

- You can use a wildcard symbol (*) to get a collection of all the tag elements in a portion of the document. For example, to get a list of all the tagged elements in the table object, you can write:

```
var tab_tags = tableObj.getElementsByTagName("*");
```

- To get the number of elements, you can use the length property:

```
<script type="text/javascript"><!--  
var divs = document.getElementsByTagName('div');  
var nr_divs = divs.length;  
--></script>
```

1. Applying a property to elements with the same tag

To apply a property to the HTML elements that have the same tag, you have to traverse the Array obtained by the `getElementsByName()` method. As a general syntax, it can be used the following code:

```
var elms = document.getElementsByTagName('tagName');  
for (var i=0; i<elms.length; i++) {  
    elms[i].property;  
}
```

- "elms[i].property" apply the "property" to all "elms" (tag elements).

Here's an example:

```
<script type="text/javascript"><!--  
function ad_style(tag) {  
    // From - http://coursesweb.net/javascript/  
    // Gets an array-object with all the tags specified in 'tag' parameter  
    var get_tags = document.getElementsByTagName(tag);  
  
    // traverse the get_tags array  
    for (var i=0; i<get_tags.length; i++) {  
        // assigns style properties  
        get_tags[i].style.textDecoration = 'underline';  
        get_tags[i].style.color = 'blue';  
    }  
}
```

```

}
--></script>
<h4 style="cursor:pointer;" onclick="ad_style('span')"><u>Click</u></h4>
Content with three SPAN tags:<br />
coursesweb.net : <span>free</span> web <span>Courses</span> and
<span>tutorials</span>.

```

- When you click on a specifically defined word (Click), the text inside each `` tag will be underlined and blue

2. getElementById() and getElementsByTagName()

When you use `"document.getElementsByTagName('tagName')"` it gets a reference to all 'tagName' in the HTML document. There are cases when you want to get only the tags within another tag. In this case, add an ID to that HTML element. In the JS script you can use **getElementById()** to get an object of that element, then, you can apply **"getElementsByTagName()"** to that object.

Syntax:

```
var variable = document.getElementById("id").getElementsByTagName("tagName");
```

Example:

```

<script type="text/javascript"><!--
// From - http://coursesweb.net/javascript/
function ad_style2(id, tag) {
    // gets the element with the ID passed to 'id' parameter
    var el_id = document.getElementById(id);

    // gets all 'tags' inside the "el_id"
    var get_tags = el_id.getElementsByTagName(tag);

    // travers the get_tags array
    for (var i=0; i<get_tags.length; i++) {
        // applies css style
        get_tags[i].style.color = 'blue';
        get_tags[i].style.fontWeight = 'bold';
    }
}
--></script>
<h4 style="cursor:pointer;" onclick="ad_style2('ul2',
'li')"><u>Click</u></h4>
<ul id="ul1">
    <li>Text within ul1 ...</li>
    <li>Text within ul1 ...</li>
    <li>Text within ul1 ...</li>
</ul><hr />
<ul id="ul2">
    <li>Text within ul2 ...</li>
    <li>Text within ul2 ...</li>
    <li>Text within ul2 ...</li>
</ul>

```

- The `"ad_style2(id, tag)"` function uses the first parameter (id) for the ID of the main element, that contains the 'tag' elements, whose name is passed in the second parameter.
- This function will set CSS style properties only to `` tags inside the `` with `id="ul2"`
- You can use the same function for other (id and tag) groups.

3. `getAttribute()` and `getElementsByTagName()`

There are cases when you want to apply JavaScript effects or properties only to some elements with the same attribute, from a group of same tags.

If you want to add a JS effect or property to HTML elements that have the same class attribute (and the same tag), use the `getAttribute('class')` method inside the "for()" instruction used to traverse the Array obtained by "`getElementsByTagName()`".

Syntax:

```
var elms = document.getElementsByTagName(tagName);
for(var i=0; i<elms.length; i++) {
    if (elms[i].getAttribute('class')== 'className') {
        // JS code ...
    }
}
```

To understand better, study the following example. It has three `` tags in a paragraph, and clicking on the word "Click", will call a JS function that adds a background color only to the SPAN tags with class = "cls".

```
<script type="text/javascript"><!--
// From - http://coursesweb.net/javascript/
function ad_style3(tag, clasa) {
    // gets the 'tag' elements
    var get_tags = document.getElementsByTagName(tag);

    // traverse the 'get_tags' array
    for (var i=0; i<get_tags.length; i++) {
        // if the element has the class fom 'clasa' parameter
        if (get_tags[i].getAttribute('class')==clasa) {
            // apply 'backgroundColor'
            get_tags[i].style.backgroundColor = '#adfead';
        }
    }
}
--></script>
<h4 style="cursor:pointer;" onclick="ad_style3('span',
'cls')"><u>Click</u></h4>
<p>Content with three SPAN tags:<br />
www.marplo.net: Web <span class="cls">Courses</span>, and <span
class="cls">tutorials</span>, all <span>free</span>.</p>
```

4. Getting the tags when the page is loaded

Adding JavaScript events (like `onclick()`, `onmouseover()`, ...) in a HTML tag, can make a function to be called when that event is triggered. But when you want to apply a JS event to multiple HTML elements, you have to add that event-attribute to each of them. This thing can agglomerate the HTML code, and it's better when you can separate the HTML and JavaScript code.

There is another way too. JavaScript is able to get the HTML elements and store events in memory, when the page is loaded, without adding event-attribute in HTML tag.

To do that, basically you have to create a function as a value of a variable (`var varName = function() { ... }`) and to call it with the `window.onload` property (`window.onload = varName;`), it makes that function to be executed when the page is loaded.

Study the code of the following example. It takes the contents and the "class" attribute value of each tag when the page is loaded. These values can be displayed in an alert window when you click on tags.

```
<html>
<head>
<title>Preloading tags</title>
<style type="text/css">
    .cli { cursor:pointer; }
</style>
<script type="text/javascript">
// creates a function as variable
var get_li = function () {
    // gets the LI tags
    var tags_li = document.getElementsByTagName('li');
    var nr_li = tags_li.length;           // gets the number of LI tags

    // traverse the "tags_li" array
    for (i=0; i<nr_li; i++) {
        // apply the 'onclick' event to each, assigning a function that must
        // be called by this event
        tags_li[i].onclick = function () {
            var clasa = this.getAttribute('class');           // gets the value
            of class attribute
            var continut = this.textContent || this.innerText; // gets
            te content
            alert('class = '+ clasa+ "\n"+ continut);           // sets the Alert
            return false;
        };
    }
};

// calls the 'get_li' variable when the page is loaded
window.onload = get_li;
</script>
</head>
<body>
<ul>
    <li class="cli">http://coursesweb.net/</li>
    <li class="cli">www.marplo.net/</li>
    <li class="cli">www.yahoo.com</li>
</ul>
</body>
</html>
```

- Notice that has not been added any "onclick" attribute, however, this event is stored in memory and when you click on each it displays an Alert with its data.